Towards a Faster Randomized Parcellation Based Inference

Andrés HOYOS-IDROBO*, Gaël VAROQUAUX*, Bertrand THIRION*,
*INRIA Parietal, Neurospin, bât 145, CEA Saclay, 91191 Gif sur Yvette, France
firstname.lastname@inria.fr

***Abstract*—In neuroimaging, multi-subject statistical analysis is an essential step, as it makes it possible to draw conclusions for the population under study. However, the lack of power in neuroimaging studies combined with the lack of stability and sensitivity of voxel-based methods may lead to non-reproducible results. A method designed to tackle this problem is Randomized Parcellation-Based Inference (RPBI), which has shown good empirical performance. Nevertheless, the use of an agglomerative clustering algorithm proposed in the initial RPBI formulation to build the parcellations entails a large computation cost. In this paper, we explore two strategies to speedup RPBI: Firstly, we use a fast clustering algorithm called Recursive Nearest Agglomeration (ReNA), to find the parcellations. Secondly, we consider the aggregation of p-values over multiple parcellations to avoid a permutation test. We evaluate their the computation time, as well as their recovery performance. As a main conclusion, we advocate the use of (permuted) RPBI with ReNA, as it yields very fast models, while keeping the performance of slower methods.**

***Index Terms*—Group analysis; reproducibility; parcellation; multiple comparisons**

I. Introduction

Statistical analyses of subjects groups are used to detect some common effects across individuals or some differences across sub-populations. The standard approach for statistical inference in neuroimaging is mass-univariate inference, where one computes a statistic in each voxel. This leads to a multiple comparison problem, given the large number of tests performed. Thus, the statistical significance of the voxel intensity test can be corrected with various statistical procedures [1].

The main issue with such analyses are the inter-subject variability of brain shape, vasculature, and function [2]. The standard approach is to register and normalize each subject into a common space. However, a perfect voxel-to-voxel correspondence is not possible. In practice, this is often mitigated by applying spatial smoothing to the data, hence increasing the overlap between subject-specific activated regions [3]. Yet, this approach strongly biases the shape of the signal of interest and thus can only be used sparingly.

One proposed approach to overcome the lack of correspondence between individual images at the voxel level is Randomized Parcellation-Based Inference (RPBI) [4]. This algorithm has shown a good empirical performance, which may be linked to its data-driven parcellation step, and semi-parametric nature. However, this comes with a large computation cost.

Our contribution: Here we propose two strategies to speedup the RPBI method. Firstly, we explore the use of a fast agglomerative clustering algorithm to build the parcellations. Secondly, we introduce another approach to aggregate p-values over different parcellations, hence avoiding a permutation test. Finally, we show the benefit of using these methods to reduce the computation time needed for statistical inference.

Notation: Vectors are written using bold lower-case, e.g. $\mathbf{x}$. Matrices are written using bold capital letters, e.g. $\mathbf{X}$.

II. Methods: Statistical modeling for group studies

We consider the images produced by an experiment (typically the brain maps displaying some combination of brain activity in response to well-chosen experimental stimuli), and aim at checking a statistical relationship between these summary images and some covariates of interest.

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{n \times p}$ is a matrix that represents the contrasts of interest measured across individuals, with one image per subject, hence n images, each one with p descriptors (e.g. voxels or parcels of an fMRI contrast image), and $\mathbf{X} \in \mathbb{R}^{n \times d_{\text{factors}}}$ is a second level design matrix that groups the explanatory variables of interest. Note that the variables in $\mathbf{X}$ can be of any type (e.g. genetic, behavioral, experimental, etc.), $\boldsymbol{\beta} \in \mathbb{R}^{d_{\text{factors}} \times p}$ denotes the population-level effects, and $\boldsymbol{\epsilon}$ is the observation noise (often considered as Gaussian). In neuroimaging, the question of interest is where in the brain a certain combination of the factors of interest yields a positive effect on average in the population, i.e. $\mathbf{c}^\mathsf{T}\boldsymbol{\beta} > 0$, where $\mathbf{c}$ is a suitable vector of contrast on the population-level effect [2]. Different types of contrasts correspond to different statistical questions. The problem boils down to estimating with which confidence one can reject the null hypothesis $\mathbf{c}^\mathsf{T}\boldsymbol{\beta} = 0$.

Variability of brain shape: The observed cross-subject variability in brain organization limits the relevance of a voxel-by-voxel description of the data, and yields a reduced sensitivity to detect the true effects.

Brain parcellations help to mitigate the alignment problems, as they take into account the spatial structure of brain images. Hence, we can reduce the dimension of the data by grouping similar neighboring voxels, moving form the voxel-space to a parcel-space. To do this, we can use anatomical/functional atlases or data-driven approaches.

A. Randomized parcellation-based inference (RPBI)

RPBI was proposed in [4] as an alternative to the Threshold-Free Cluster Enhancement (TFCE) [10]. It finds the consensus of the statistical decision over multiple parcellations of the brain volume. The method works as follows:

1) **Build brain parcellations:** RPBI uses data resampling (bootstrap) to find B parcellations. Each parcellation is built using a Ward agglomerative clustering algorithm [5] on the resampled data.
2) **Perform standard analysis:** Statistical analysis is performed on each parcellation, yielding one statistical value per parcel, further binarized by comparison to a statistical threshold t.
3) **Compute the RPBI statistics:** We form the voxel-based sum of binary variables testing obtained at the previous step: note that each voxel inherits the values of the parcels it was assigned to.
4) **Permutation test:** RPBI statistics are compared to their permutation distribution to control the family-wise error rate (FWER) at the level of voxels.

Formally, let $\mathcal{C}$ be the set of parcellations, and $\mathcal{V}$ be the set of voxels under consideration. Given a voxel v and a parcellation C, the parcel-based thresholding function θ_t is defined as:

$$\theta_t(v, C) = \begin{cases} 1 & F(\Phi_C(v)) > t, \\ 0 & \text{otherwise.} \end{cases}, \tag{2}$$

where $\Phi_C : V \to C$ is a mapping function that associates each voxel v with a parcel from the parcellation C. For a predefined test, F returns the F-statistic associated with the average signal of a given parcels (other statistics are also possible). Finally, the aggregating statistic at a voxel v is given by the counting function K_t:

$$K_t(v, \mathcal{C}) = \sum_{C \in \mathcal{C}} \theta_t(v, C). \tag{3}$$

$K_t(v, \mathcal{C})$ represents the number of times a voxel v was part of a parcel associated with a statistical value larger than t across the folds of the analysis conducted on the set of parcellations $\mathcal{C}$. The parameter t is arbitrary, but it is typically set to ensure Bonferroni-corrected control at 0.1 the parcel-level analyzes.

B. P-value aggregation

There are different schemes to combine non-independent p-values. They can be organized in two categories: quantile combination methods and order statistic methods. Both approaches rely on the fact that under the null hypothesis a p-value from an absolutely continuous test statistic has a uniform distribution from zero to one [6]. Under this assumption, these methods can control the FWER.

In [7], Meinshausen et al. propose a way to aggregate p-values in high dimensional regression settings. This approach relies on splitting the data into two non-overlapping sets, namely the train and test sets. On the train set, one uses a sparse coding algorithm to select the support of active voxels. Then, the test set is used to find the corresponding p-values. Bootstrap resampling is then used to remove the dependency on the splitting. Finally, the different p-values are aggregated via quantile combination.

III. Speeding-up RPBI

A. Using a fast agglomerative clustering

In practice, most of the computation time of RPBI is related to the construction of the parcellations. To tackle this, we propose to use the recursive nearest agglomeration algorithm (ReNA) [8]. This is an agglomerative clustering algorithm that finds clusters in linear-time. ReNA has shown similar performance to Ward in a larger number of clustering settings with impressive speedups. To build the clusters, ReNA relies on extracting the connected components of a 1-nearest-neighbor (1-NN) graph. To reach the desired number k of clusters, it is applied recursively. The algorithm outline is as follows:

1) **To build the graph representation:** one uses a weighted adjacency matrix. The non-zero weights of this matrix encode the topology of brain images, and the values denote the similarity between the nodes. Note that at the first iteration, the nodes correspond to the voxels.
2) **To build the clusters:** one has to find the connected components of the adjacency matrix.
3) **To reduce the graph:** this is done by removing edges of the adjacency matrix, and replacing values of the nodes by the average of the connected components.

Finally, one repeats the previous steps until reaching the desired number k of clusters.

B. P-value aggregation over multiple parcellations

Our second approach to reduce the computation time of RPBI consists in adapting the p-values aggregation in high-dimensional regression to handle parcellations. First, we apply ReNA on bootstrapped subsamples to generate B randomized parcellations, each one with k parcels. Then for $b = 1, \ldots, B$:

1) Perform the statistical test at parcel-level.
2) Then, define the adjusted[1] (non-aggregated) p-values as

$$P^{(b)}_{\text{adj},j} \leftarrow \min(P^{(b)}_j k, 1), \quad j = 1, \ldots, p. \tag{4}$$

Finally, a p-value for each predictor $j = 1, \ldots, p$ is given by the γ-corrected empirical γ-quantile function, for any fixed $0 < \gamma < 1$. This is defined as

$$P_j \leftarrow \min\left(1, q_\gamma\left(\left\{P^{(b)}_{\text{adj},j}/\gamma, \quad b = 1 \ldots, B\right\}\right)\right), \tag{5}$$

where $q_\gamma(.)$ is the (empirical) γ-quantile function. For $\gamma = 0.5$ this estimators boils down to twice the median. Yet, in a different context [9], the authors use the median to aggregate p-values.

[1] This corresponds to a parcel-level Bonferroni correction.

IV. Experiments: Empirical verification

We first checked that the error rate is controlled at the nominal level for all methods, which is indeed the case. In this section, we investigate the computation time and recovery performance of various inference methods. We compare the execution time and recovery of *i)* voxel-level group analysis via ordinary least squares (OLS), which is the standard method in neuroimaging; *ii)* threshold-free cluster enhancement (TFCE) [10]; *iii*) RPBI with Ward [4]; *iv)* RPBI with ReNA, and *v)* ReNA aggregation. When clustering is applied, we set the number k of clusters to 5% of the number p of voxels[2]. We set the number B of bootstrap replications to 100. Additionally, for the OLS, RPBI, and TFCE we perform 10 000 permutation tests to control the FWER. We set $\gamma = 0.5$ for the ReNA aggregation. We use Nilearn [11] to handle neuroimaging data. We rely on ReNA, presented in [8]. We use FSL [12] for the TFCE algorithm.

A. Datasets

Brainomics/localizer dataset: We use the functional Magnetic Resonance Imaging (fMRI) data from the *functional localizer* dataset [13]. It contains data from 94 participants. The primary goal of this dataset is to map basic brain networks, opening up a dataset of healthy subjects for neuroscientific studies. All the images were processed using SPM8. We use for the analysis a *calculation versus sentences* contrast.

Human Connectome Project (HCP) [14]: We consider the HCP (500 release) fMRI language task. This dataset contains 500 participants (13 removed for quality reasons). The primary goal of this dataset is network discovery, which is facilitated by probing experimental task paradigms that are known to tap on well characterized neural networks [15]. We profited from the HCP "minimally preprocessed" pipeline [16]. We use for the analysis the language processing (semantic and phonological processing), and the emotion processing protocol.

Pseudo-ground truth: We define as ground truth the thresholded p-value map to keep the 5% of the most active voxels ($p \ll 1 \times 10^{-6}$). Since we use a voxel-based threshold, the ground truth may be biased to voxel-based procedures (thus disadvantaging our method).

We did not use additional smoothing on any dataset.

B. Benchmark of analysis methods: computation time

First, we analyze the computation time of various group-analysis methods. To build the confidence intervals, we perform the analysis on 10 subsamples of 20 subjects each.

Fig. 1 shows the comparison of the computation time of various inference algorithms. OLS is overall the fastest, with a computation time 8 times smaller than the mean across methods. It is followed by ReNA-aggregation. TFCE and RPBI with ReNA display the same performance ($p < 0.05$ paired Wilcoxon rank test). RPBI with Ward's clustering is the slowest with a computation time 3 times greater than the mean across methods. Table I gives a summary of the absolute wall clock for each method.

[2]We consider a useful dimension reduction range, $k \in [\frac{p}{20}, \frac{p}{10}]$. This regime gives a good trade-off between computational efficiency and data fidelity [8].

Method	Datasets	
	Localizer	HCP
OLS (permuted)	36 sec	3 min 21 sec
TFCE	7 min 6 sec	23 min 20 sec
RPBI Ward	10 min 30 sec	1 h 21 min 30 sec
ReNA + aggregation	44 sec	15 min 34 sec
RPBI ReNA	3 min 36 sec	40 min 19 sec

TABLE I

Computation time of various analysis methods: The fastest method is OLS (permuted), followed by the ReNA aggregation. TFCE and RPBI with ReNA have a performance that varies across datasets. RPBI with Ward is consistently the slowest. Note that the computation time is obtained using a single CPU.

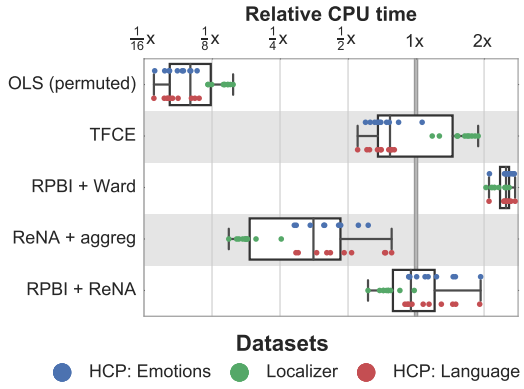

Fig. 1. **Comparison of the computation time of various analysis methods:** Relative computation time for different datasets. The values are displayed relative to the mean over all methods. OLS (permuted) is the fastest, followed by ReNA aggregation. TFCE and RPBI with ReNA have the same performance, and display an intermediate computation time. The classical RPBI with Ward's clustering is the slowest, by a factor of 3.

C. Recovery

Now, we investigate the performance of several group-analysis methods to retrieve the reference of the activity pattern of the population. To do so, we randomly drew 20 subjects and perform our experiment on 10 such different subsamples. Because of the reduced number of subjects used, we cannot expect to retrieve the same activation map as in the pseudo-ground truth (the full-sample analysis) due to a loss in statistical power. To estimate the recovery, precision-recall curves are constructed by reporting the proportion of true positives in the detections (precision) for different levels of recovery of the ground truth (recall).

Fig. 3 shows that ReNA aggregation has a slightly better performance than OLS. TFCE fails to recover when the threshold is liberal, possibly due to its non-local nature. The same behavior is observed for RPBI on the HCP dataset. Fig. 2 shows an illustration of activation patterns obtained via various inference algorithms. TCFE and aggregation-based methods display a higher sensitivity with respect to OLS. We can see that the activations from TFCE and RPBI maps are similar and wider than others methods, whereas OLS is much more conservative. ReNA aggregation displays an intermediate behavior.

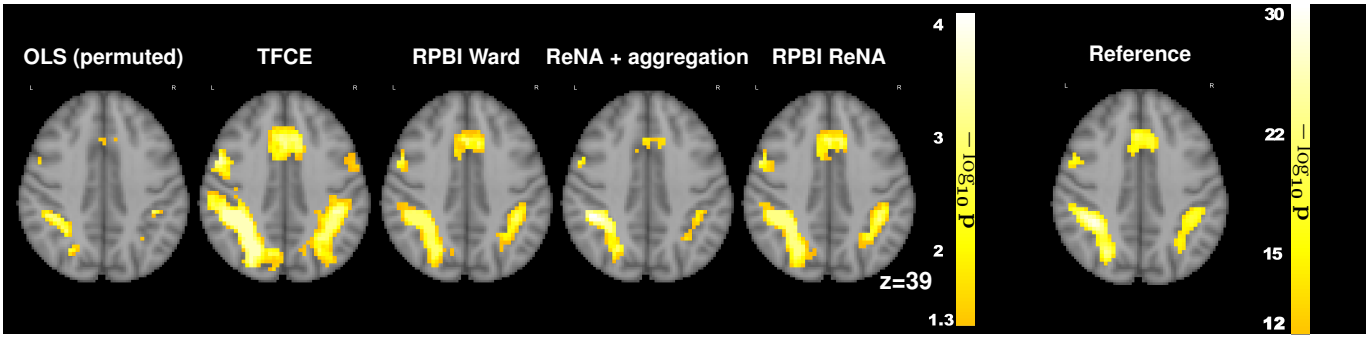

Fig. 2. **Qualitative comparison of the results obtained with various analysis methods:** One-sample test in one subgroup of subjects. The displayed maps correspond to the negative log p-value associated with a non-zero intercept test on *calculations vs sentences* fMRI contrast from the localizer dataset. The subgroup maps are thresholded at $-\log_{10} p = 1.3$ FWER corrected, and the reference at $-\log_{10} p = 12$. The permuted OLS displays smaller supra-threshold clusters. ReNA aggregation detects more voxels than OLS, yet is more conservative than RPBI and TFCE. RPBI method with ReNA and Ward's clustering find activation patterns that are similar to the ground truth.

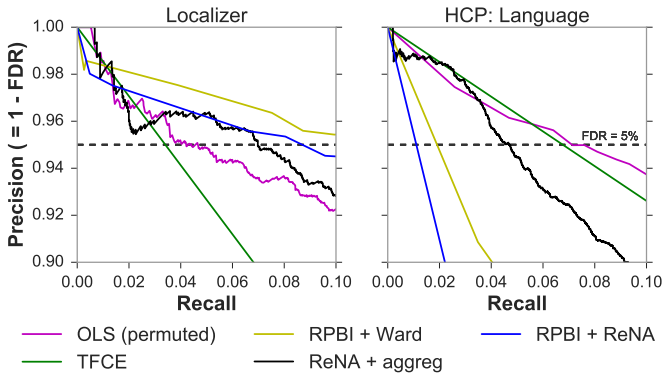

Fig. 3. **Recovery of various analysis methods:** Precision-recall curves across 10 subsamples containing 20 subsamples. This curve is built by thresholding the reference map at several arbitrary levels.

V. Discussion: Use fast clustering

Our validation over several datasets (two shown here) indicate that ReNA aggregation slightly improves the sensitivity over OLS. Yet, in most of the experiments RPBI and TFCE display better performance. Regarding the computation time, the use of ReNA consistently yields faster inference algorithms. It reduces the computation time of RPBI by a factor 3 with respect to Ward, while keeping a similar performance. The ReNA aggregation displays a conservative behavior. Nevertheless, we think that the aggregation over multiple parcellations is a direction of research that needs further investigation, and can lead to a better understanding of RPBI-like algorithms.

Acknowledgment: This project received funding from the European Union's Horizon 2020 Framework Program for Research and Innovation under Grant Agreement No 720270 (Human Brain Project SGA1).

References

[1] R. A. Poldrack, J. A. Mumford, and T. E. Nichols, *Handbook of functional MRI data analysis*. Cambridge University Press, 2011.

[2] B. Thirion, "Functional neuroimaging group studies," 2016.

[3] A. F. Sol, S. chung Ngan, G. Sapiro, X. Hu, and A. Lpez, "Anisotropic 2d and 3d averaging of fmri signals," *IEEE Trans. on Medical Imaging*, vol. 20, pp. 86–93, 2001.

[4] B. Da Mota, V. Fritsch, G. Varoquaux, T. Banaschewski, G. J. Barker, A. L. Bokde, U. Bromberg, P. Conrod, J. Gallinat, H. Garavan *et al.*, "Randomized parcellation based inference," *NeuroImage*, vol. 89, pp. 203–215, 2014.

[5] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, p. 236, 1963.

[6] T. M. Loughin, "A systematic comparison of methods for combining p-values from independent tests," *Computational statistics & data analysis*, vol. 47, no. 3, pp. 467–485, 2004.

[7] N. Meinshausen, L. Meier, and P. Bühlmann, "P-values for high-dimensional regression," *Journal of the American Statistical Association*, vol. 104, no. 488, pp. 1671–1681, 2009.

[8] A. Hoyos-Idrobo, G. Varoquaux, J. Kahn, and B. Thirion, "Recursive nearest agglomeration (ReNA): fast clustering for approximation of structured signals," *arXiv:1609.04608*, 2016.

[9] M. A. van de Wiel, J. Berkhof, and W. N. van Wieringen, "Testing the prediction error difference between 2 predictors," *Biostatistics*, vol. 10, no. 3, pp. 550–560, 2009.

[10] S. M. Smith and T. E. Nichols, "Threshold-free cluster enhancement: Addressing problems of smoothing, threshold dependence and localisation in cluster inference," *NeuroImage*, vol. 44, pp. 83 – 98, 2009.

[11] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Muller, J. Kossaifi, A. Gramfort, B. Thirion, and G. Varoquaux, "Machine learning for neuroimaging with scikit-learn," *Frontiers in neuroinformatics*, vol. 8, 2014.

[12] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney *et al.*, "Advances in functional and structural mr image analysis and implementation as fsl," *Neuroimage*, vol. 23, pp. S208–S219, 2004.

[13] D. P. Orfanos, V. Michel, Y. Schwartz, P. Pinel, A. Moreno, D. Le Bihan, and V. Frouin, "The brainomics/localizer database," *NeuroImage*, 2015.

[14] D. V. Essen, K. Ugurbil, E. Auerbach, D. Barch, T. Behrens, R. Bucholz, A. Chang, L. Chen, M. Corbetta, S. Curtiss, S. D. Penna, D. Feinberg, M. Glasser, N. Harel, A. Heath, L. Larson-Prior, D. Marcus, G. Michalareas, S. Moeller, R. Oostenveld, S. Petersen, F. Prior, B. Schlaggar, S. Smith, A. Snyder, J. Xu, and E. Yacoub, "The human connectome project: A data acquisition perspective," *NeuroImage*, vol. 62, pp. 2222–2231, 2012.

[15] D. M. Barch, G. C. Burgess, M. P. Harms, S. E. Petersen, B. L. Schlaggar, M. Corbetta, M. F. Glasser, S. Curtiss, S. Dixit, C. Feldt, D. Nolan, E. Bryant, T. Hartley, O. Footer, J. M. Bjork, R. Poldrack, S. Smith, H. Johansen-Berg, A. Z. Snyder, D. C. V. Essen, and W. U.-M. H. Consortium, "Function in the human connectome: task-fmri and individual differences in behavior." *Neuroimage*, vol. 80, 2013.

[16] M. F. Glasser, S. N. Sotiropoulos, J. A. Wilson, T. S. Coalson, B. Fischl, J. L. Andersson, J. Xu, S. Jbabdi, M. Webster, and J. R. Polimeni, "The minimal preprocessing pipelines for the human connectome project," *Neuroimage*, vol. 80, 2013.